A Noise-Tolerant Hybrid Model of a Global and a Local Learning Module

Natsuki Oka and **Kunio Yoshida**
Information & Network Research Laboratory
Matsushita Research Institute Tokyo, Inc.
3-10-1, Higashimita, Tama-ku, Kawasaki 214, Japan
Email: {oka,yoshida}@mrit.mei.co.jp

Abstract

Proposed is GLLL2, a hybrid architecture of a global and a local learning module, which learns default and exceptional knowledge respectively from noisy examples. The global learning module, which is a feedforward neural network, captures global trends gradually, while the local learning module stores local exceptions quickly. The latter module distinguishes noise from exceptions, and learns only exceptions, which ability makes GLLL2 noise-tolerant. The results of experiments show the process in which training examples are formed into default and exceptional knowledge, and demonstrate that the predictive accuracy, the space efficiency, and the training efficiency of GLLL2 is higher than those of each individual module.

Introduction

In cognitive science, whether learners induce rules or remember exemplars has been at issue recently. In particular, learning processes of quasi-regular tasks that involve both regularities and exceptions has attracted considerable attention.

We developed GLLL1, a hybrid architecture of a global and a local learning module. We made experiments on it, confirmed its high accuracy and efficiency (Oka & Yoshida 1992), and found (Oka & Yoshida 1993): 1) regular and irregular examples are distinguished and learned separately by the two learning modules; 2) learning progresses in three stages, and overgeneralization occurs in the second stage; 3) outputs for highly unusual inputs are produced as if they are members of regular examples. These findings fit qualitatively human learning data reported by Marcus et al. (Marcus et al. 1990) and Pinker (Pinker 1991).

Although GLLL1 performs well if it is given noise-free examples, its performance degrades rapidly with increasing noise rate in training examples. This is because instance-based learning algorithms, one of which the local learning module uses, are sensitive to noise in nature, and the influence of noise is emphasized when the local learning module is made to learn irregular examples selectively. In this paper, we therefore propose GLLL2, a noise-tolerant extension of GLLL1.

We begin by describing the structure and the algorithm of GLLL2 in the second section. Then, experimental results in the third section demonstrate the process in which training examples are formed into default and exceptional knowledge, and high accuracy and efficiency of GLLL2. The fourth section contains comparisons with related work, and the final section is conclusion.

GLLL2: A Noise-Tolerant Hybrid Model

GLLL2 learns to guess an appropriate output for an input, given training examples. The architecture consists of two learning modules: a global learning module (G-module) and a noise-tolerant local learning module (NTL-module), two example selection modules which cause the two learning modules to be trained over different portions of the examples, and an output selection module for producing the final output (Figure 1).

We use a back-propagation network (Rumelhart, Hinton & Williams 1986) as the G-module, and a noise-tolerant extension of RCE network (Reilly, Cooper & Elbaum 1982) as the NTL-module. The back-propagation network and the RCE network have complementary strengths. The former captures global trends of a functional form gradually, while the latter is good at filling in local exceptions quickly.

The RCE network is a type of instance-based classifiers, and stores selectively training examples with their hyperspherical influence regions. Given a new input, an output is produced according to stored examples whose influence regions include the input. If the output is wrong, the new input is stored and/or the influence regions are narrowed.

In order to make the overall system noise-tolerant, the NTL-module must be able to distinguish noise from exceptions. In order to realize this, the learning algorithm of the RCE network is extended in four respects:

1. It maintains the numbers of the correct and the incorrect uses of each saved example. It also maintains the distribution of correctly and incorrectly classified inputs in the influence region of each saved example.
2. The influence region of a saved example should not be narrowed when incorrect uses for the example is

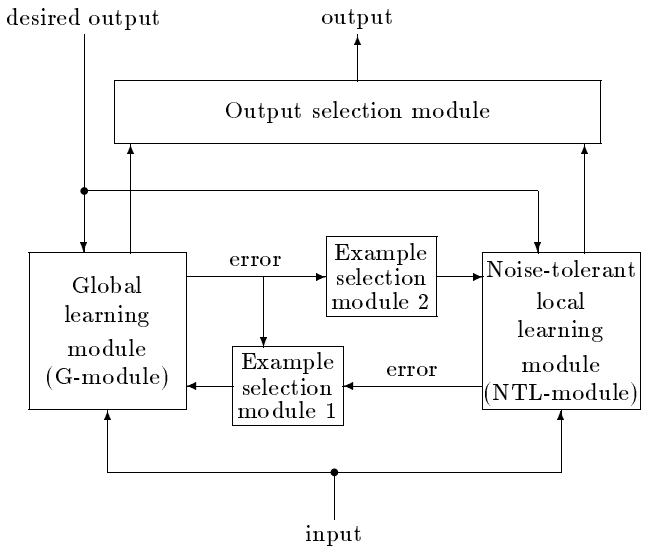

Figure 1: The structure of GLLL2. GLLL2 has two complementary learning modules that interact through two example selection modules. An output selection module produces the final output.

likely to be caused solely by noise (case I), but should be narrowed when the stored example itself is likely to be produced by noise, or when the influence region is likely to contain different class regions (case II). The NTL-module distinguishes the two cases on the following criterion: if the rate of the correct uses of a saved example is lower than a predetermined threshold θ, or the distribution of incorrectly classified inputs in the influence region of the example is not uniform, the influence region of the example is narrowed; otherwise the influence region is not narrowed.

If the noise rate R, the rate of incorrectly labeled training examples, is known beforehand, the threshold θ can be set at a slightly smaller value than R, in which case we need not consider the distribution in order to distinguish the two cases. However, the noise rate is usually unknown in realistic problems, and hence θ should be set at a sufficiently small value in order to avoid unnecessarily narrowing influence regions, while overly small θ makes true class boundaries vague if we do not consider the distribution condition, that is, the addition of the distribution condition to the criterion enables the NTL-module to distinguish exceptions from noise. To sum up, the NTL-module distinguishes exceptions from noise based on the distribution of incorrectly classified inputs in the influence region of each stored example.

3. Given a new input, among statistically reliable saved examples, one that has the lowest potential decides the output of the NTL-module. When there is no statistically reliable stored example that has nonzero potential, the output is defined as indefinite. The potential is given by

$$p_e(i) = \begin{cases} -(c_e - 0.5) \times \frac{r_e - d_{ie}}{r_e} & \text{if } c_e > 0.5 \wedge r_e > d_{ie} \\ 0 & \text{otherwise} \end{cases}$$

where $p_e(i)$ is the potential of a stored example e at the point of a given input i, c_e is the rate of correct uses of the stored example, d_{ie} is the distance between the point of the input and the site of the stored example, and r_e is the radius of the influence region of the stored example.

4. Each stored example is accompanied with its impression degree which increases if the example is used correctly, and decreases according to the lapse of time. Stored examples are discarded if their impression degrees become less than a threshold, that is, examples of low utility such as noisy examples or rare examples are dropped.

The two example selection modules cause the two learning modules to be trained over different portions of the examples as follows: example selection module 1 receives output errors from the NTL-module and the G-module, and makes the G-module selectively learn examples for which the outputs of the NTL-module are not correct or examples for which the output errors of the G-modules are small; conversely, example selection module 2 receives output errors from the G-module, and makes the NTL-module forget or not store examples for which the output errors of the G-module become small enough.

The NTL-module stores training examples with their hyperspherical influence regions, and thereby it not only learns input-output relations of the training examples, but also learns whether it can reliably estimate output to a given input, that is, if the given input comes within at least one influence region of the stored examples, it produces definite and reliable output, otherwise its output is indefinite.

This characteristic feature of the NTL-module makes the output selection algorithm very simple, that is, given a new input, if the output of the NTL-module is definite, the output selection module selects the output of the NTL-module as the final output, otherwise it selects the output of the G-module. To sum up, the NTL-module also serves as a learner for the output selection, which double function enables incremental and efficient learning.

Experimental Results

Experiment 1: Formation of Default and Exceptional Knowledge

In this section, we demonstrate the learning process in GLLL2. In order to visualize the learning process, we use a two dimensional region identification task. As shown in the following figures, a waved dotted line

from the lower left corner to the upper right corner divides a square region into two regions **B** (black) and **W** (white), and there is an exceptional circular region **W** in the upper left part. The task is to predict the label of the region from the x and y coordinates in the plane. Randomly generated training examples were given incrementally. There is no repetitive presentation. The noise rate was set to 5%, that is, the outputs of randomly selected 5 percent examples were reversed.

The NTL-module roughly covers the whole input space in the first stage (Figure 2), because the learning speed of the NTL-module is much faster than that of the G-module.

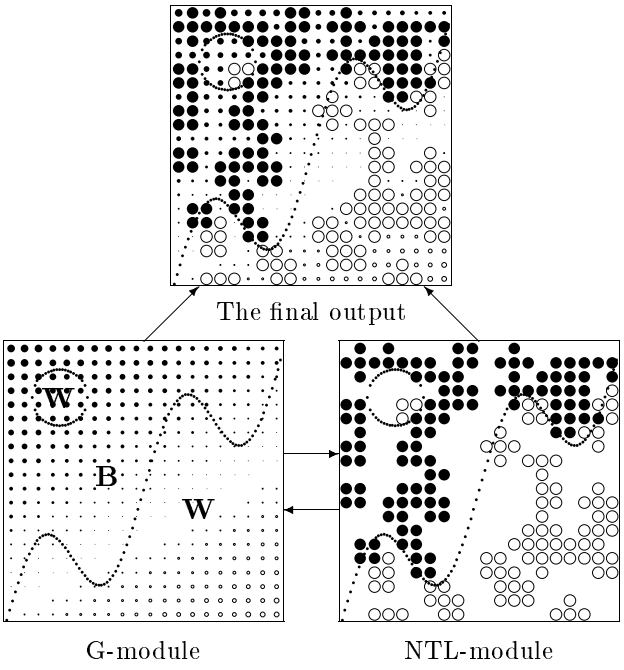

Figure 2: The output maps of the global learning module (G-module), the noise-tolerant local learning module (NTL-module), and the total system after the presentation of 300 examples in a noisy region identification task. The Dotted lines represent boundaries between regions **B** (black) and **W** (white). Black and white circles at regular intervals indicate the outputs **B** and **W** respectively at sample points. In the lower left map, the sizes of the circles are proportional to the G-module's output values, and in the lower right map, no circle denote indefinite output. The NTL-module roughly covers the whole input space in this stage, because learning in the G-module is not enough yet.

The G-module has a tendency to capture global trends of a functional form gradually. It becomes, as a result, being able to produce correct outputs to regular examples in later stages (Figure 3). On the other hand, the NTL-module is made to forget the examples which are learned by the G-module. Consequently, with the progress of learning in the G-module, the NTL-module becomes in charge mainly of irregular examples (Figure 3). If the NTL-module's output is definite, it becomes the final output, otherwise the G-module's output becomes the final output, that is, the G-module works as default (Figure 3).

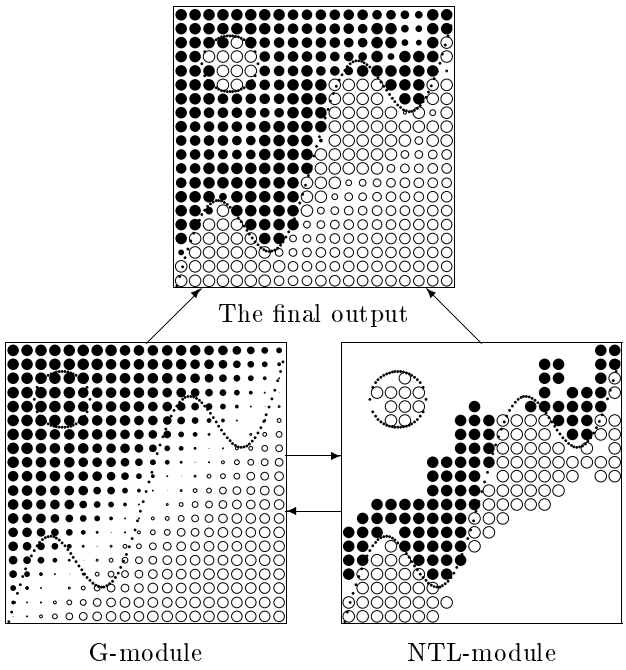

Figure 3: The output maps of the G-module, the NTL-module, and the total system after the presentation of 3000 examples in the same task. The G-module learns to produce correct outputs to regular examples gradually. With the progress of learning in the G-module, the NTL-module is made to be in charge mainly of irregular examples. If the NTL-module's output is definite, it becomes the final output, otherwise the G-module's output becomes the final output. The sharing of the examples between the two learning modules is successful in spite of the existence of noise.

Experiment 2: Accuracy and Efficiency

In this section, we compare the accuracy and the efficiency of GLLL2 with those of each individual module. Here we use another two dimensional region identification task which has regions of different shape, and has nonuniform example distribution. As shown in Figure 4, a diagonal line divides a square region into two triangular regions **B** (black) and **W** (white), and there are four exceptional square regions which are marked with **B** or **W**. In the four square regions drawn with thick lines, examples were generated four times as frequent as in the other regions. The noise rate was set to 5% again.

Figure 5 shows the accuracy and the efficiency of

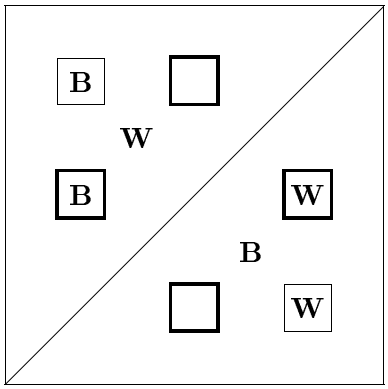

Figure 4: The region identification task used in the second experiment. It contains exceptional examples and noise. Distribution of examples is not uniform.

GLLL2, the individual NTL-module, and the individual G-module in the second task. The horizontal axis indicates the size of the models, that is, the number of the stored examples for the NTL-module, the number of the hidden units for the G-module, and the total of the both numbers for GLLL2. The vertical axis indicates the output accuracy for unknown test inputs. The results of two trials per size setting are plotted for each model.

Randomly generated 6000 examples constitute a training example set. The accuracies of GLLL2 and of the individual NTL-module were computed after a single presentation of the training example set, while accuracy of the individual G-module was tested after 100 presentations because the learning speed of the G-module is much slower than that of the NTL-module or GLLL2. The accuracy also depends on the setting of adjustable parameters the models have. A nearly best setting for each condition was selected by trial and error. The result of this experiment demonstrates clearly that the predictive accuracy, the space efficiency, and the training efficiency of GLLL2 is higher than those of each individual module.

Related Work

Learning Regular and Irregular Examples with Single-Architecture Models

Rumelhart et al. (Rumelhart & McClelland, 1986), and Plunkett et al. (Plunkett & Marchman 1989) insisted that the processing of regular and exceptional items can coexist within a single back-propagation network. Kruschke (Kruschke 1991) proposed a model of human category learning based on exemplar-based representation. Plaut et al. (Plaut & McClelland 1993) showed that an attractor network can be trained to perform quasi-regular tasks.

Although their models succeeded in learning regular and irregular examples, the result of experiment 2 in

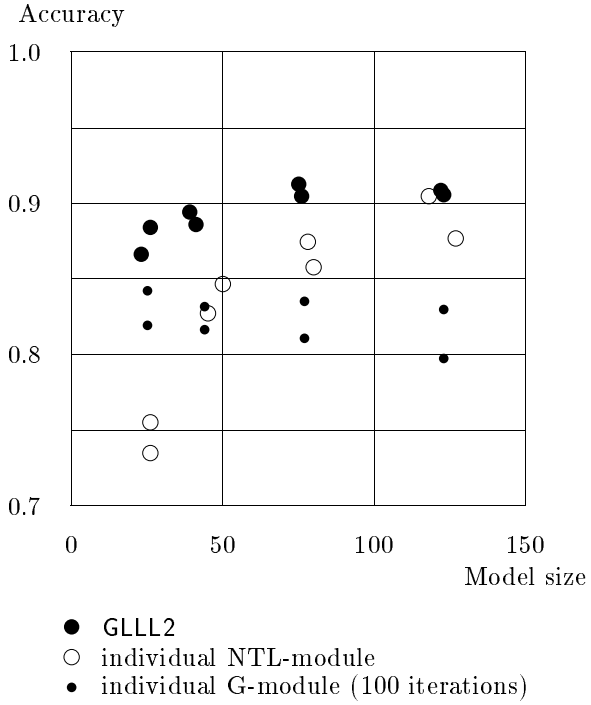

Figure 5: The predictive accuracy, the space efficiency, and the training efficiency of GLLL2 is higher than those of each individual module. The accuracies of GLLL2 and of the individual NTL-module were computed after a single presentation of a training example set, while the accuracy of the individual G-module was tested after 100 presentations because the learning speed of the G-module is much slower than that of the NTL-module or GLLL2.

the previous section indicates that single-architecture models tend to be poor in accuracy and efficiency.

Modularized Models

Pinker (Pinker 1991) insisted that the dissociations between regular and irregular inflection are evidence for the existence of separate neural subsystems for regular and exceptional items respectively. McClelland et al. (McClelland, McNaughton & O'Reilly 1994) also suggest that the neocortex may be optimized for the gradual discovery of the shared structure of events and experiences, and that the hippocampal system is there to provide a mechanism for rapid acquisition of new information without interference with previously discovered regularities.

Golding et al. (Golding & Rosenbloom 1991) combined a case-based reasoning system that memorizes exceptions, and a rule-based system that works as defaults. Rules are given beforehand in their system, while rules are implicitly learned in our system.

Wolpert (Wolpert 1992) and Zhang et al. (Zhang, Mesirov & Waltz 1992) proposed learning systems

composed of separate expert modules and a combiner. First, each expert module independently learns, then the combiner learns to produce the final output from the outputs of the expert modules. Because of this two-stage learning procedure, their systems may have higher accuracy than GLLL2. However, their learning algorithms are not incremental, while our algorithm is incremental and much more efficient. Lee et al. (Lee & Wang 1993) also proposed a neural network classifier by combining hyperplane with exemplar approach that are not incremental either.

Jacobs et al. (Jacobs & Jordan 1993) proposed a learning system composed of a share network, many expert networks, and a gating network. The modular architecture with a share network is a system that dedicates the share network to learning the common features of a set of tasks and dedicates other expert networks to learning the features that are unique to each task. Kruschke et al. (Kruschke & Erickson 1994) also proposed a combination of a global and a local learning module using the competitive gating mechanisms. They need to specify the number of expert networks or the number of exemplar nodes beforehand in their systems, while we need not do so in GLLL2.

French (French 1995) proposed an interactive tandem-network architecture, consists of two continually interacting networks, one – the LTM network – dynamically storing prototypes of the patterns learned, the other – the STM network – being responsible for short-term learning of new patterns. We believe that our architecture is more appropriate for learning exceptions than his architecture, because his architecture does not use a network of local representation.

Noise-Tolerant Instance-Based Learning

Since instance-based learning algorithms are highly sensitive to noisy training examples, several noise-tolerant extensions have been proposed.

Aha and Kibler's extension (Aha & Kibler 1989) requires evidence that saved examples be significantly good classifiers before it allows them to be used for subsequent classification tasks, and it discards those saved examples that appear to be noisy (i.e., those examples whose classification performance is poor after several classification attempts). However, their noise-tolerating extension has several limitations: 1) it has no means of distinguishing exceptions from noise, 2) it has a side effect of making true boundaries vague, and 3) the criterion they use to drop noisy examples is based on class frequency, which seems to be not appropriate, adequate examples are once saved.

Okamoto's noise-tolerant extension (Okamoto 1990) decides whether to shrink the influence region of a saved example or not in accordance with the classification record for the example, discards those saved examples whose influence region becomes small, and among saved examples, one that has the smallest value of $d_{ie} - r_e$ decides the output, where d_{ie} is the distance between the point of an input i and the site of a stored example e, and r_e is the radius of the influence region of the stored example. His extension also has several limitations: 1) it has no means of distinguishing exceptions from noise, 2) it has a side effect of making true boundaries vague, 3) the criterion he uses to drop examples is not appropriate, if the distribution of examples is not uniform, and 4) the criterion for deciding outputs is not appropriate because examples with small influence regions are too lightly treated.

Cost et al.'s algorithm (Cost & Salzberg 1993) attaches weights to saved exemplars according to their performance history: reliable exemplars are given smaller weights, making them appear closer to a new example. This method is effective in dealing with exceptional instances to a certain extent, whereas it is not appropriate for ignoring noisy instances.

Scofield et al.'s extension (Scofield et al. 1988) employs square-well potential whose depth is proportional to the frequency of correct uses of a exemplar. This method aims at learning the probability distributions of non-separable point sets, and hence it is not suitable for dealing with noise and exceptions.

In comparison with these extensions, our extension has a characteristic feature that distinguishes exceptions from noise based on the distribution of incorrectly classified inputs in the influence region of each stored example.

Conclusion

We proposed GLLL2, a noise-tolerant hybrid architecture of a global and a local learning module, which learns default and exceptional knowledge respectively from examples. We explained the process in which training examples are formed into default and exceptional knowledge. The results of experiments demonstrated that the predictive accuracy, the space efficiency, and the training efficiency of GLLL2 is higher than those of each individual module.

Currently we are making experiments on the effect of the second and the third respects of the noise-tolerant extension of the NTL-module, which respects were described in the second section. Preliminary results show that in most cases, each of the second and the third respects of the extension is effective for improving the predictive accuracy, and that there are some cases in which the employment of the both respects has an additive effect.

We are going to apply GLLL2 to real world applications such as CAD systems, and we have a further plan to combine GLLL2 with a rule-based reasoning system (Oka 1991).

References

Aha, D. W., and Kibler, D. 1989. Noise-Tolerant Instance-Based Learning Algorithms. In Proceedings of IJCAI-89, 794–799.

Cost, S., and Salzberg, S. 1993. A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning*, 10:57–78.

French, R. M. 1995. Interactive Tandem Networks and the Problem of Sequential Learning. In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society. Lawrence Erlbaum Associates.

Golding, A. R., and Rosenbloom, P. S. 1991. Improving Rule-Based Systems through Case-Based Reasoning. In Proceedings of AAAI-91, 22–27.

Jacobs, R. A., and Jordan, M. I. 1993. Learning Piecewise Control Strategies in a Modular Neural Network Architecture. *IEEE Transactions on Systems, Man, and Cybernetics* 23(2):337–345.

Kruschke, J. K. 1991. ALCOVE: A Connectionist Model of Human Category Learning. In R. P. Lippmann et al. eds., *Advances in Neural Information Processing Systems 3*, 649–655. Morgan Kaufmann.

Kruschke, J. K., and Erickson, M. A. 1994. Learning of Rules That Have High-Frequency Exceptions: New Empirical Data and a Hybrid Connectionist Model. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society. Lawrence Erlbaum Associates.

Lee, H.-M., and Wang, W.-T. 1993. Training of a Neural Network Classifier by Combining Hyperplane with Exemplar Approach. In Proceedings of the IEEE International Conference on Neural Networks, 494–499.

Marcus, G. F.; Ullman, M.; Pinker, S.; Hollander, M.; Rosen, T. J.; and Xu F. 1990. Overregularization, Center for Cognitive Science Occasional Paper # 41, Massachusetts Institute of Technology.

McClelland, J. L.; McNaughton, B. L.; and O'Reilly, R. C. 1994. Why there are Complementary Learning Systems in the Hippocampus and Neocortex: Insights from the Successes and Failures of Connectionist Models of Learning and Memory, Technical Report, PDP.CNS.94.1, Carnegie Mellon University.

Oka, N. 1991. Hybrid Cognitive Model of Conscious Level Processing and Unconscious Level Processing. In Proceedings of 1991 IEEE International Joint Conference on Neural Networks, 485–490.

Oka, N., and Yoshida, K. 1992. Combining Back-Propagation and Memory-Based Learning. In Proceedings of International Symposium on Neural Information Processing, 54–57. Kyushu Institute of Technology, Iizuka, Fukuoka, Japan.

Oka, N., and Yoshida, K. 1993. Learning Regular and Irregular Examples Separately. In Proceedings of 1993 IEEE International Joint Conference on Neural Networks, 171–174.

Okamoto, Y. 1990. Neural Network Model for Realtime Adaptation to Rapidly Changing Environment. *The Transactions of the Institute of Electronics, Information and Communication Engineers* J73-D-II(8): 1186-1191, (in Japanese).

Pinker, S. 1991. Rules of Language, *Science* 253: 530–535.

Plaut, D. C., and McClelland, J. L. 1993. Generalization with Componential Attractors: Word and Nonwords Reading in an Attractor Network. In Proceedings of the 15th Annual Conference of the Cognitive Science Society.

Plunkett, K., and Marchman, V. 1989. Pattern Association in a Back Propagation Network: Implications for Child Language Acquisition, Technical Report, # 8902, Center for Research in Language, University of California, San Diego.

Reilly, D. L.; Cooper, L. N.; and Elbaum, C. 1982. A Neural Model Category Learning, *Biol. Cybern.* 45:35–41.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning Internal Representations by Error Propagation. In D. E. Rumelhart et al. eds., *Parallel Distributed Processing 1*, 318–362. MIT Press.

Rumelhart, D. E., and McClelland, J. L. 1986. On Learning the Past Tense of English Verbs. In J. L. McClelland et al. eds., *Parallel Distributed Processing 2*, 216–271. MIT Press.

Scofield, C. L.; Reilly, D. L.; Elbaum, C.; and Cooper, L. N. 1988. Pattern Class Degeneracy in an Unrestricted Storage Density Memory, In D. Z. Anderson ed., *Neural Information Processing Systems*, 674–682. New York: American Institute of Physics.

Wolpert, D. H. 1992. Stacked Generalization, *Neural Networks* 5:241–259.

Zhang, X.; Mesirov, J. P.; and Waltz, D. L. 1992. Hybrid System for Protein Secondary Structure Prediction, *J. Mol. Biol.* 225:1049–1063.